Computer Science

On the Performance of Spectral Graph Partitioning Methods

Stephen Guattery Gary L. Miller

December 1994

CMU-CS-94-228

Carnegie Mellon

On the Performance of Spectral Graph Partitioning Methods

Stephen Guattery Gary L. Miller

December 1994

CMU-CS-94-228

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

To appear in the Sixth Annual ACM/SIAM Symposium on Discrete Algorithms (SODA '95)

This work was supported in part by NSF Grant CCR-9016641.

This work was also supported in part by the Air Force Materiel Command (AFMC) and the Advanced Research Projects Agency (ARPA) under contract number F19628-93-C-0193. In addition, IBM, Motorola, and the NSF/Presidential Young Investigator Award under Grant No. CCR-8858087, TRW, and the U.S. Postal Service gave their support.

Keywords: Algorithms, Graph Theory, Graph Algorithms, Numerical Linear Algebra, Eigenvalues

Abstract

Computing graph separators is an important step in many graph algorithms. A popular technique for finding separators involves spectral methods. However, there is not much theoretical analysis of the quality of the separators produced by this technique; instead it is usually claimed that spectral methods "work well in practice." We present an initial attempt at such analysis. In particular, we consider two popular spectral separator algorithms, and provide counterexamples that show these algorithms perform poorly on certain graphs. We also consider a generalized definition of spectral methods that allows the use of some specified number of the eigenvectors corresponding to the smallest eigenvalues of the Laplacian matrix of a graph, and show that if such algorithms use a constant number of eigenvectors, then there are graphs for which they do no better than using only the second smallest eigenvector. Further, when applied to these graphs the algorithm based on the second smallest eigenvector performs poorly with respect to theoretical bounds. Even if an algorithm meeting the generalized definition uses up to n^ϵ for $0 < \epsilon < \frac{1}{4}$ eigenvectors, there exist graphs for which the algorithm fails to find a separator with a cut quotient within $n^{\frac{1}{4}-\epsilon} - 1$ of the isoperimetric number. We also introduce some facts about the structure of eigenvectors of certain types of Laplacian and symmetric matrices; these facts provide the basis for the analysis of the counterexamples.

1 Introduction

Spectral methods (i.e., methods using the eigenvalues and eigenvectors of a matrix representation of a graph) are widely used to compute graph separators. Typically, the Laplacian matrix representation B of a graph G is used; the Laplacian B of a graph G on n vertices is the $n \times n$ matrix with the degrees of the vertices of G on the diagonal, and entry $b_{ij} = -1$ if G has the edge (v_i, v_j) and 0 otherwise. The eigenvector $\mathbf{u}_2$ corresponding to λ_2 (the second-smallest eigenvalue of B) is computed, and the vertices of the graph are partitioned according to the values of their corresponding entries in $\mathbf{u}_2$ [PSL90, HK92]. The goal is to compute a small separator; that is, as few edges or vertices as possible should be deleted from the graph to achieve the partition.

Although spectral methods are popular, there is little theoretical analysis of how well they do in producing small separators. Instead, it is usually claimed that such methods "work well in practice," and tables of results for specific examples are often included in papers (see e.g. [PSL90]). Thus there is little guidance for someone wishing to compute separators as to whether or not this technique is appropriate. Ideally, practitioners should have a characterization of classes of graphs for which spectral separator techniques work well; this characterization might be in terms of how far the computed separators can be from optimal. As a first step in this direction, we consider two spectral separation algorithms that partition the vertices on the basis of the values of their corresponding entries in $\mathbf{u}_2$, and provide counterexamples for which each of the algorithms produces poor separators. We further consider a generalized definition of spectral methods that allows the use of more than one of the eigenvectors corresponding to the smallest non-zero eigenvalues, and show that there are graphs for which any such algorithm does poorly.

The first algorithm is a popular technique that consists of bisecting a graph by partitioning the vertices into two equal-sized sets based on each vertex's entry in the eigenvector corresponding to the second-smallest eigenvalue. A graph in our first counterexample class looks like a ladder with the top $2/3$ of its rungs kicked out (see Figure 1); a straightforward spectral bisection algorithm cuts the remaining rungs, whereas the optimal bisection is made by cutting across the ladder above the remaining rungs. (We refer to this graph as the "roach graph" because its outline looks roughly like the body of a cockroach and two long antennae.)

It is possible to modify the spectral bisection algorithm in ways that allow it to generate a better separator for the roach graph. Some examples of possible modifications are presented in [HK92]; these examples still use a partition based on $\mathbf{u}_2$. We consider a simple spectral separator algorithm, the "best threshold cut" algorithm, based on the most general of these suggested modifications, and present a second class of graphs that defeats this algorithm. This class of graphs consists of crossproducts of path graphs and graphs consisting of a pair of complete binary trees connected by an edge between their roots.

Finally, we consider a more general definition of purely spectral separator algorithms that subsumes the two preceding algorithms. This general definition allows the use of some specified number of eigenvectors corresponding to the smallest eigenvalues of the Laplacian of a graph. For any such algorithm that uses a fixed number of eigenvectors we show there are graphs for which it does no better than using the "best threshold cut" algorithm. Further, the separator produced when the "best threshold cut" algorithm is applied to these graphs is as bad as possible (to within a constant) with respect to theoretical bounds on the size of the separators produced.

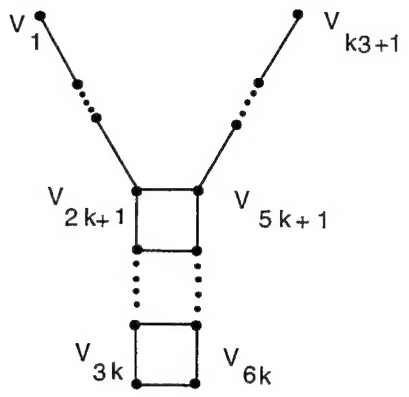

Figure 1: The Roach Graph

We also show that if a purely spectral algorithm uses up to n^ϵ eigenvectors for $0 < \epsilon < \frac{1}{4}$, there exist graphs for which the algorithm fails to find a separator with a **cut quotient** (i.e., the ratio between the number of edges cut and the size of the smaller set in the vertex partition) within $n^{\frac{1}{4}-\epsilon} - 1$ of the optimum (the optimum cut quotient is called the *isoperimetric number*). We also note that the counterexample graphs can be extended to graphs that could conceivably be used as three-dimensional finite-element meshes – that is, graphs that could be encountered in practice.

This paper makes one additional contribution: Our counterexamples have simple structures and intuitively would be expected to cause problems for spectral separator algorithms. The challenge is to provide good bounds on λ_2 for these graphs. For this purpose we have developed theorems about the spectra of graphs with particular symmetries that exist in our counterexamples.

Specifics are given in the text that follows. Section 2 gives some history of spectral methods and details of the algorithms we discuss in this paper. Section 3 gives the graph and matrix terminology that we will use, and presents some useful facts about Laplacian matrices. Section 4 gives the counterexample for the spectral bisection algorithm; Section 5 gives the counterexample for the "best threshold cut" algorithm. Finally, Section 6 discusses our generalized notion of spectral separator algorithms, and shows that there are graphs for which any such algorithm performs poorly.

2 Spectral Methods for Computing Separators

The roots of spectral partitioning go back to Hoffmann and Donath [DH73], who proved a lower bound on the size of the minimum bisection of a graph, and Fiedler [Fie73][Fie75], who explored the properties of λ_2 and its associated eigenvector for the Laplacian of a graph. There has been much subsequent work, including Barnes's partitioning algorithm [Bar82], Boppana's work that included a stronger lower bound on the minimum bisection size [Bop87], and the particular bisection and graph partitioning problems that we are considering in this paper [HK92] [PSL90] [Sim91]. (We note that spectral methods have not been limited to graph partitioning; work has been done using the spectrum of the adjacency matrix in graph coloring [AG84] and using the Laplacian spectrum to prove theorems about expander graph and superconcentrator properties [AM85] [Alo86] [AGM87].

The work on expanders has explored the relationship of λ_2 to the isoperimetric number; Mohar has given an upper bound on the isoperimetric number using a strong discrete version of the Cheeger inequality [Moh89]. Reference [CDS79] is a book-length treatment of graph spectra, and it predates many of the results cited above.)

A basic way of computing a graph bisection using spectral information is presented in [PSL90]. We will refer to this algorithm as **spectral bisection**. Spectral bisection works as follows:

- Represent G by its Laplacian B, and compute $\mathbf{u}_2$, the eigenvector corresponding to λ_2 of B.
- Assign each vertex the value of its corresponding entry in $\mathbf{u}_2$. This is the **characteristic valuation** of G.
- Compute the median of the elements of $\mathbf{u}_2$. Partition the vertices of G as follows: the vertices whose values are less than or equal to the median element form one part of the partition; the rest of the vertices form the other part. The set of all edges between the two parts forms an edge separator.
- If a vertex separator is desired, it can be computed from the edge separator as described in the next section.

Since the graph bisection problem is NP-complete [GJ79], spectral bisection may not give an optimum result. That is, spectral bisection is a heuristic method. A number of modifications have been proposed that may improve on its performance. These modified heuristics can give splits other than bisections. In such cases, we will use the cut quotient or separator ratio (the **separator ratio** is the ratio between the number of edges cut and the product of the sizes of the two sets in the vertex partition) in judging how close the split is to optimal. Computing the separator with the minimum ratio is NP-hard (see, e.g., [LR88]). The following modifications, all of which use the characteristic valuation, are presented in [HK92]:

- Partition the vertices based on the signs of their values;
- Look for a large gap in the sorted list of eigenvector components, and partition the vertices according to whether their values are above or below the gap; and
- Sort the vertices according to value. For each index $1 \leq i \leq n-1$, consider the ratio for the separator produced by splitting the vertices into those with sorted index $\leq i$ and those with sorted index $> i$. Choose the split that provides the best separator ratio.

Note that the last idea subsumes the first two. We will consider a variant of this algorithm below. Since this algorithm does not consider the case where multiple vertices may have the same value, we will specify that only splits between vertices with different values are considered (we refer to such cuts as **threshold cuts**). We call this algorithm the **"best threshold cut" algorithm**; the fact that we've slightly changed the definition above does not alter its performance with respect to the counterexamples below (other than slightly simplifying the analysis).

Also note that the idea of cutting at an arbitrary point along the sorted order can be extended to choosing two split points, where the corresponding partitions are the vertices with values between the split points, and those with values above the upper or below the lower split point. Again, the pair yielding the best ratio is chosen.

The algorithms mentioned so far have only used the eigenvector $\mathbf{u}_2$. Another possibility is to look at the partitions generated by the set of eigenvectors for some number of smallest eigenvalues: for each vertex, a value would be assigned by computing a function of that vertex's eigenvector components. Partitions could then be generated in the same way as they are for $\mathbf{u}_2$ in the various algorithms given above.

Given the variety of heuristics cited above, it would be nice to know which ones work well for which classes of graphs. It would be particularly useful if it were possible to state reasonable bounds on the performance of these heuristics for classes of graphs commonly used in practice (e.g., planar graphs, planar graphs of bounded degree, three-dimensional finite element meshes, etc.). Unfortunately, this is not the case. We start by proving that spectral bisection may produce a bad separator for a bounded-degree planar graph in Section 4; first, however, we need to introduce some terminology and background results.

3 Terminology, Notation, and Background Results

We assume that the reader is familiar with the basic definitions of graph theory (in particular, for undirected graphs), and with the basic definitions and results of matrix theory. A graph consists of a set of vertices V and a set of edges E; we denote the vertices (respectively edges) of a particular graph G as $V(G)$ (respectively $E(G)$) if there is any ambiguity about which graph is referred to. The notation $|G|$ will be sometimes be used as a shorthand for $|V(G)|$. When it is clear which graph we are referring to, we will use n to denote $|V|$.

We use capital letters to represent matrices and bold lower-case letters for vectors. For a matrix A, a_{ij} or $[A]_{ij}$ represents the element in row i and column j; for the vector $\mathbf{x}$, x_i or $[\mathbf{x}]_i$ represents the i^{th} entry in the vector. The notation with square brackets is useful in cases where adding subscripts to lower-case letters would be awkward (e.g., where the matrix or vector name is already subscripted). The notation $\mathbf{x} = 0$ indicates that all entries of the vector $\mathbf{x}$ are zero; $\vec{1}$ indicates the vector that has 1 for every entry. For ease of reference, we will index the eigenvalues of an $n \times n$ matrix in non-decreasing order. λ_1 represents the smallest eigenvalue, and λ_n the largest. For $1 < i < n$, we have $\lambda_{i-1} \leq \lambda_i \leq \lambda_{i+1}$. We use the notation $\lambda_i(A)$ (respectively $\lambda_i(G)$) to indicate the i^{th} eigenvalue of matrix A (respectively of the Laplacian of graph G) if there is any ambiguity about which matrix (respectively graph) the eigenvalue belongs to. $\mathbf{u}_i$ represents the eigenvector corresponding to λ_i.

We use the term **path graph** for a tree that has exactly two vertices of degree one. That is, a path graph is a graph consisting of exactly its maximal path.

The **crossproduct** of two graphs G and H (denoted $G \times H$) is a graph on the vertex set $\{(u, v) \mid u \in V(G), v \in V(H)\}$, with $((u, v), (u', v'))$ in the edge set if and only if either $u = u'$ and $(v, v') \in E(H)$ or $v = v'$ and $(u, u') \in E(G)$. It is easy to see that $G \times H$ and $H \times G$ are isomorphic. One way to think of a graph crossproduct is as follows: Replace every vertex in G with a copy of H. Each edge e in G is then replaced by $|H|$ edges, one between each pair of corresponding vertices in the copies of H that have replaced the endpoints of e. That is, for each edge (v_i, v_j) in G, there is a copy i and a copy j of H. For each vertex u in H, add an edge between vertex u in copy i and the vertex u in copy j. An example is shown in the figure below.

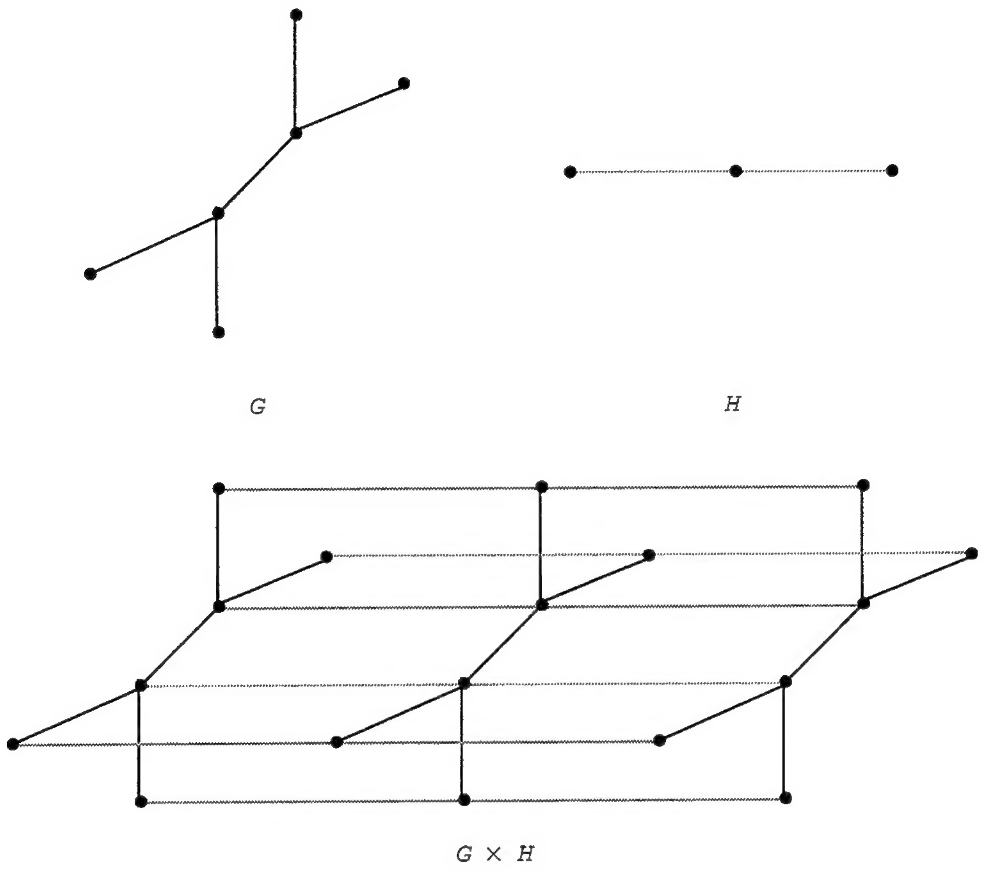

Figure 2: A Graph Crossproduct Example

For a connected graph G, an **edge separator** is a set S of edges that, if removed, would break the graph into two (not necessarily connected) components G_1 and G_2 such that there are no edges between G_1 and G_2. (We will assume that an edge separator is a minimal set with respect to the particular G_1 and G_2.) A **vertex separator** is a set S of vertices such that if these vertices and all incident edges are removed, the graph is broken into two components G_1 and G_2 such that there are no edges between G_1 and G_2 (again, we'll assume that such a separator is minimal). The goal in finding separators is to find a small separator that breaks the graph into two fairly large pieces; often this notion of "large pieces" is expressed as a restriction that the number of vertices in either component be at least some specified fraction of the number of vertices in G. For edge separators, this can be stated more generally in terms of the **separator ratio** ρ, defined as $|S|/\left(|G_1| \cdot |G_2|\right)$. The **optimum ratio separator** ρ_{opt} is the one that minimizes the separator ratio over all separators for a particular graph [LR88].

A related concept (again, for edge separators) is the **isoperimetric number** $i(G)$, defined as:

$$\min_S \left(\frac{|S|}{\min(|G_1|, |G_2|)} \right).$$

We will refer to the quantity $|S| / \min(|G_1|, |G_2|)$ as the **cut quotient** for the edge separator S. It is easy to see that $n\rho_{opt} > i(G) \geq n\rho_{opt}/2$. As noted in the Introduction above, finding a cut with the minimum separator ratio or with a cut quotient equal to the isoperimetric number is NP-hard.

It is well known that an edge separator S can easily be converted into a vertex separator S' by considering the bipartite graph induced by S (where the parts of the bipartition are determined by the components G_1 and G_2), and setting S' to be a minimum edge cover for that graph.

Graphs can be represented by matrices. For example, the **adjacency matrix** A of a graph G is defined as $a_{ij} = 1$ if and only if $(v_i, v_j) \in E(G)$; $a_{ij} = 0$ otherwise. (For such representations we will assume that the vertices have been numbered to correspond to the indices used in the matrix.)

A common matrix representation of graphs is the **Laplacian**. Let D be the matrix with $d_{ii} = \text{degree}(v_i)$ for $v_i \in V(G)$, and all off-diagonal entries equal to zero. Let A be the adjacency matrix for G as defined above. Then the Laplacian representation of G is the matrix $B = D - A$.

The following are some useful facts about the Laplacian matrix:

- The Laplacian is symmetric positive semidefinite, so all its eigenvalues are greater than or equal to 0 (see e.g. [Moh88]).
- A graph G is connected if and only if 0 is a simple eigenvalue of the Laplacian of G (see e.g. [Moh88]).
- The following characterization of λ_2 holds (see e.g. [Fie73]):

$$\lambda_2 = \min_{\mathbf{x} \perp \vec{1}} \frac{\mathbf{x}^T A \mathbf{x}}{\mathbf{x}^T \mathbf{x}}.$$

- If G is a crossproduct of two graphs G_1 and G_2, then the eigenvalues of the Laplacian of G are all pairwise sums of the eigenvalues of G_1 and G_2 (see e.g. [Moh88]).
- For any vector $\mathbf{x}$ and Laplacian matrix B of the graph G, we have (see e.g. [HK92]):

$$\mathbf{x}^T B \mathbf{x} = \sum_{(v_i, v_j) \in E(G)} (x_i - x_j)^2 \qquad (1)$$

- For a graph G and its Laplacian B, λ_2 of B figures in upper and lower bounds on the isoperimetric number for G [Moh89]. In particular, if G is not one of K_1, K_2, or K_3, we have:

$$\frac{\lambda_2}{2} \leq i(G) \leq \sqrt{\lambda_2(2\Delta - \lambda_2)}, \qquad (2)$$

  where Δ is the maximum degree of any vertex in G. This implies the size of any bisection is at least $\frac{n\lambda_2}{4}$.

The proof of the upper bound in (2) has interesting implications about the threshold cuts based on the second eigenvector. For any connected graph G, consider the characteristic valuation. The vertices of G will receive $k \leq n$ distinct values; let $t_1 > t_2 > \ldots > t_k$ be these values. For each threshold $t_i,\ i < k$, divide the vertices into those with values greater than t_i, and those with values less than or equal to t_i. Compute the cut quotient q_i for each such cut, and let q_{min} be the minimum over all q_i's. The following theorem can be derived from the proof of Theorem 4.2 in [Moh89]:
Theorem 3.1 *Let G be a connected graph with maximal vertex degree Δ with second smallest eigenvalue λ_2. Further, let G not be any of K_1, K_2, or K_3. Let q_{min} be as defined above. Then*

$$\frac{\lambda_2}{2} \leq q_{min} \leq \sqrt{\lambda_2(2\Delta - \lambda_2)}.$$

This can be extended to the separator ratio for the best $\mathbf{u}_2$ cut in the obvious way.

A **weighted** graph is a graph for which a real value w_i is associated with each vertex v_i, and a real, nonzero weight w_{ij} is associated with each edge (v_i, v_j) (we consider a zero edge weight to indicate the lack of an edge). For our analysis, we need a matrix representation for weighted graphs. Fiedler extended the notion of the Laplacian to graphs with positive edge weights [Fie75]; he referred to this representation as the **generalized Laplacian**. However, we need an even more general representation that can be used for any weighted graph. Hence we define the **standard matrix representation** B of a weighted graph G as follows: B has $b_{ii} = w_i$; for $i \neq j$ and $(v_i, v_j) \in E(G)$, $b_{ij} = -w_{ij}$, and $b_{ij} = 0$ otherwise. Note that the Laplacian matrix of a graph is also the standard matrix representation of the graph with vertex weights set to be the degrees of the vertices, and all edge weights set to 1.

Note that the standard matrix representation of any weighted graph is a real symmetric matrix, and that any such matrix can be represented as a specific weighted graph. Any theorems about symmetric, real matrices apply to standard matrix representations of graphs. The following two theorems about the interlacing of eigenvalues are particularly useful when applied to standard matrix representations.

The **First Interlacing Property** (the following statement is from page 411 of reference [GL89]; the proof is in [Wil65]): If A_r denotes the leading $r \times r$ principal submatrix of an $n \times n$ symmetric matrix A, then for $r = 1 : n-1$ the following interlacing property holds:

$$\begin{aligned}\lambda_{r+1}(A_{r+1}) \geq \lambda_r(A_r) \geq \lambda_r(A_{r+1}) \geq \ldots \\ \ldots \geq \lambda_2(A_{r+1}) \geq \lambda_1(A_r) \geq \lambda_1(A_{r+1}).\end{aligned}$$

The **Second Interlacing Property** (the following statement is a restricted version of a theorem from page 412 of reference [GL89]; the proof is in [Wil65]): Suppose $B = A + \alpha \mathbf{c}\mathbf{c}^T$, where A is a real symmetric $n \times n$ matrix, $\mathbf{c}$ is a real unit-length vector, and α is real and greater than 0. Then for all $i,\ 1 \leq i \leq n-1$ we have:

$$\lambda_i(A) \leq \lambda_i(B) \leq \lambda_{i+1}(A)$$

The Second Interlacing Property implies that if an edge e is added to a graph G to produce the graph G', then $\lambda_2(G) \leq \lambda_2(G')$.

3.1 The Structure of Laplacian and Standard Matrix Representation Eigenvectors

The theorems and lemmas presented in this section are useful in proving results about the eigenvectors of the families of graphs presented in later sections. The proofs of some of these results are long and technical; a reader who is interested only in understanding the counterexamples and their implications presented later in this report is advised to look at the theorem statements and examples, and to skip the proofs (although the rules for constructing odd and even components at the end of Theorem 3.4 may be of interest to some readers).

The first set of results concern eigenvalues of Laplacian matrices of graphs with automorphisms of order 2. A **graph automorphism** is a permutation ϕ on the vertices of the graph G such that $(v_i, v_j) \in E(G)$ if and only if $(v_{\phi(i)}, v_{\phi(j)}) \in E(G)$. The **order** of a graph automorphism is the order of the permutation on the vertices.

For weighted graphs, we add two conditions to the definition of automorphism: the weights of vertices v_i and $v_{\phi(i)}$ must be equal for all i, and the weights of edges (v_i, v_j) and $(v_{\phi(i)}, v_{\phi(j)})$ must be equal.

The next two theorems concern the structure of eigenvectors with respect to automorphisms of order 2. They hold both for Laplacians of graphs under the standard definition of automorphism, and for standard matrix representations of weighted graphs under the definition of automorphisms for weighted graphs.

Let G be a graph with an automorphism ϕ of order 2. Let B be the Laplacian of G. A vector $\mathbf{x}$ that has $x_i = x_{\phi(i)}$ for all i in the range $1 \leq i \leq n$ is an **even** vector with respect to the automorphism ϕ; an **odd** vector $\mathbf{y}$ has $y_i = -y_{\phi(i)}$ for all i. It is easy to show that for any even vector $\mathbf{x}$ and odd vector $\mathbf{y}$ (both with respect to ϕ), $\mathbf{x}$ and $\mathbf{y}$ are orthogonal.

Theorem 3.2 [Even-Odd Eigenvector Theorem] *Let B be the Laplacian of a graph G that has an automorphism ϕ of order 2. Then there exists a complete set $\mathcal{U}$ of orthogonal eigenvectors of B such that any eigenvector $\mathbf{u} \in \mathcal{U}$ is either even or odd with respect to ϕ. This also holds if G is a weighted graph, B the standard matrix representation of G, and ϕ a weighted graph automorphism of order 2.*

Proof: Let P be the permutation matrix that corresponds to the automorphism ϕ. Then $P^T BP = B$. If $\mathbf{u}$ is an eigenvector of B with eigenvalue λ, then so is $P\mathbf{u}$. We have the following by the definition of automorphism:

$$\left(P^T BP\right) \mathbf{u} = B\mathbf{u} = \lambda \mathbf{u}.$$

Since the automorphism is of order 2, $PP = I$. Multiplying each side of the equation above by P thus gives

$$B\left(P\mathbf{u}\right) = P\left(\lambda \mathbf{u}\right) = \lambda \left(P\mathbf{u}\right).$$

For an even vector $\mathbf{x}$, $P\mathbf{x} = \mathbf{x}$; for an odd vector $\mathbf{y}$, $P\mathbf{y} = -\mathbf{y}$.

P allows us to uniquely decompose any vector $\mathbf{x}$ into an odd component $\mathbf{x}_{odd}$ and an even component $\mathbf{x}_{even}$ as follows:

$$\mathbf{x}_{odd} = \frac{\mathbf{x} - P\mathbf{x}}{2}, \text{ and } \mathbf{x}_{even} = \frac{\mathbf{x} + P\mathbf{x}}{2}.$$

For any non-zero $\mathbf{x}$, at least one of the even or odd parts must also be non-zero.

Let $\mathcal{U}'$ be any complete set of eigenvectors of B. For an eigenvector $\mathbf{u} \in \mathcal{U}'$, it is easy to see that a non-zero even or odd component will be an eigenvector for the same eigenvalue. Since $\mathbf{u}_{odd} + \mathbf{u}_{even} = \mathbf{u}$, the set of odd and even eigenvectors resulting from decomposing all the eigenvectors spans the same space as $\mathcal{U}$. This implies the claimed result.

The proof generalizes to weighted graphs.

□

Corollary 3.3 *Let B be the standard matrix representation of a weighted graph G that has one or more automorphisms of order 2. Then the eigenvector for any simple eigenvalue is either even or odd with respect to every such automorphism.*

Proof: Let $\mathbf{u}$ be the eigenvector for some simple eigenvalue λ. Consider the decomposition of $\mathbf{u}$ into odd and even parts with respect to some automorphism ϕ with order 2. If both parts were non-zero, they would be orthogonal and eigenvectors for λ. Therefore either the odd part or the even part must be zero.

□

Since Laplacians can be considered as standard matrix representations given the right weight assignments, the preceding result also holds for Laplacians.

Theorem 3.4 [Even-Odd Graph Decomposition Theorem] *For every weighted graph G with standard matrix representation B and an automorphism ϕ of order 2, there exist two smaller weighted graphs G_{odd} and G_{even} (the odd and even components respectively) such that the eigenvalues of B_{odd} (the standard matrix representation of G_{odd}) are odd eigenvalues of B, the eigenvalues of B_{even} (the standard matrix representation of G_{even}) are even eigenvalues of B, and $|V(G_{odd})| + |V(G_{even})| = |V(G)|$. Further, a complete set of eigenvectors of B can be constructed from the eigenvectors of B_{odd} and B_{even} in a straightforward way.*

Proof: We will demonstrate a similarity transform that when applied to B gives a reducible matrix with two blocks corresponding to B_{odd} and B_{even}. First we need to introduce some notation.

The vertices of G can be divided into two disjoint sets on the basis of how ϕ operates on them. Let V_f be the set of vertices v_i such that $\phi(i) = i$ (i.e., the vertices fixed by ϕ); and let V_m be the set of vertices v_j such that $\phi(j) \neq j$ (i.e., the vertices moved by ϕ). V_m consists of vertices in orbits of length 2. We call a subset of V_m that consists of exactly one vertex from each such orbit a **representative set** and denote it V_r. In the rest of the proof we will assume that we have arbitrarily specified a particular V_r. We use n_f, n_m, and n_r respectively to denote the number of vertices in each of these sets.

Without loss of generality, we can assume that the vertices have the following numbering: the vertices in V_f are numbered 1 through n_f; the vertices in V_r are numbered from $n_f + 1$ to $n_f + n_r$. If $v_i \in V_r$, then we set $\phi(i) = i + n_r$; that is, the vertices in $V_m \setminus V_r$ are numbered $n_f + n_r + 1$ to n in the same order as the vertices in V_r with which they share orbits. Using this ordering and the definition of the automorphism, we can write B in the following block form

$$B = \begin{bmatrix} F & E_{fr} & E_{fr} \\ E_{fr}^T & R & E_{r\phi(r)} \\ E_{fr}^T & E_{r\phi(r)} & R \end{bmatrix},$$

where

- F is an $n_f \times n_f$ submatrix containing the diagonal entries for the vertices in V_f and the entries for edges between pairs of vertices in V_f;

- R is an $n_r \times n_r$ submatrix containing the diagonal entries for the vertices in V_r and the entries for edges between pairs of vertices in V_r (recall that the definition of an automorphism and the numbering we've used together imply that the same submatrix applies for the vertices in $V_m \setminus V_r$);

- E_{fr} contains the entries of B for edges between vertices in V_f and V_r (our numbering of the vertices plus the automorphism condition that (v_i, v_j) is an edge of G if and only if $(v_{\phi(i)}, v_{\phi(j)})$ is also an edge imply that the submatrix with entries for edges between V_f and $V_m \setminus V_r$ will be the same); and

- $E_{r\phi(r)}$ contains the entries of B for edges between vertices in V_r and $V_f \setminus V_r$ (again, our vertex numbering, the automorphism condition that (v_i, v_j) is an edge of G if and only if $(v_{\phi(i)}, v_{\phi(j)})$ also is an edge, and the symmetry of B imply that $E_{r\phi(r)} = E^T_{r\phi(r)} = E_{\phi(r)r}$).

We now define the orthogonal matrix T that we will use to transform B. T has the following form:

$$T = \begin{bmatrix} I_{n_f} & 0 & 0 \\ 0 & \frac{1}{\sqrt{(2)}} I_{n_r} & \frac{1}{\sqrt{(2)}} I_{n_r} \\ 0 & \frac{1}{\sqrt{(2)}} I_{n_r} & \frac{-1}{\sqrt{(2)}} I_{n_r} \end{bmatrix},$$

where the I's are identity matrices with the dimension specified in the subscript. We can transform B as follows:

$$\begin{aligned} B' &= T^T B T \\ &= \begin{bmatrix} I_{n_f} & 0 & 0 \\ 0 & \frac{1}{\sqrt{(2)}} I_{n_r} & \frac{1}{\sqrt{(2)}} I_{n_r} \\ 0 & \frac{1}{\sqrt{(2)}} I_{n_r} & \frac{-1}{\sqrt{(2)}} I_{n_r} \end{bmatrix} \begin{bmatrix} F & E_{fr} & E_{fr} \\ E^T_{fr} & R & E_{r\phi(r)} \\ E^T_{fr} & E_{r\phi(r)} & R \end{bmatrix} \begin{bmatrix} I_{n_f} & 0 & 0 \\ 0 & \frac{1}{\sqrt{(2)}} I_{n_r} & \frac{1}{\sqrt{(2)}} I_{n_r} \\ 0 & \frac{1}{\sqrt{(2)}} I_{n_r} & \frac{-1}{\sqrt{(2)}} I_{n_r} \end{bmatrix} \\ &= \begin{bmatrix} I_{n_f} & 0 & 0 \\ 0 & \frac{1}{\sqrt{(2)}} I_{n_r} & \frac{1}{\sqrt{(2)}} I_{n_r} \\ 0 & \frac{1}{\sqrt{(2)}} I_{n_r} & \frac{-1}{\sqrt{(2)}} I_{n_r} \end{bmatrix} \begin{bmatrix} F & \sqrt{2} E_{fr} & 0 \\ E^T_{fr} & \frac{1}{\sqrt{(2)}}(R + E_{r\phi(r)}) & \frac{1}{\sqrt{(2)}}(R - E_{r\phi(r)}) \\ E^T_{fr} & \frac{1}{\sqrt{(2)}}(R + E_{r\phi(r)}) & \frac{1}{\sqrt{(2)}}(-R + E_{r\phi(r)}) \end{bmatrix} \\ &= \begin{bmatrix} F & \sqrt{2} E_{fr} & 0 \\ \sqrt{2} E^T_{fr} & R + E_{r\phi(r)} & 0 \\ 0 & 0 & R - E_{r\phi(r)} \end{bmatrix}. \end{aligned}$$

Note that the resulting matrix is reducible. That is, when viewed as a weighted graph, that graph has two components. We will show that the blocks of this matrix correspond to B_{even} and B_{odd} as follows:

$$B_{even} = \begin{bmatrix} F & \sqrt{2} E_{fr} \\ \sqrt{2} E^T_{fr} & R \end{bmatrix} \text{ and } B_{odd} = R - E_{r\phi(r)}.$$

Since B' is similar to B, it has a number of useful properties. Because B' is reducible, every eigenvalue of B_{odd} is an eigenvalue B'; likewise every eigenvalue of B_{even} is an eigenvalue B'. By similarity, they are also eigenvalues of B.

Now consider an eigenvector $\mathbf{u}$ of B_{even}. Define $\mathbf{v}$ as follows: for $1 \leq i \leq n_f + n_r$ let $v_i = u_i$; let $v_i = 0$ otherwise. $\mathbf{v}$ is obviously an eigenvector of B'. We can use the matrix T to transform $\mathbf{v}$ into an eigenvector $\mathbf{w}$ of B:

$$\mathbf{w} = T\mathbf{v} = \begin{bmatrix} I_{n_f} & 0 & 0 \\ 0 & \frac{1}{\sqrt(2)} I_{n_r} & \frac{1}{\sqrt(2)} I_{n_r} \\ 0 & \frac{1}{\sqrt(2)} I_{n_r} & \frac{-1}{\sqrt(2)} I_{n_r} \end{bmatrix} \begin{bmatrix} \mathbf{v}_f \\ \mathbf{v}_r \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{v}_f \\ \frac{1}{\sqrt(2)} \mathbf{v}_r \\ \frac{1}{\sqrt(2)} \mathbf{v}_r \end{bmatrix}$$

By our numbering, it is easy to see this is an even eigenvector of B. Since $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$ all have the same eigenvalue λ, the claim about eigenvalues of B_{even} corresponding to even eigenvalues of B holds. It is easy to show that if two eigenvectors $\mathbf{u_1}$ and $\mathbf{u_2}$ of B_{even} are orthogonal, then the corresponding eigenvectors $\mathbf{w_1}$ and $\mathbf{w_2}$ are also orthogonal. Thus if an eigenvalue of B_{even} has multiplicity 2, there are two orthogonal even eigenvectors of B with that eigenvalue.

Now consider an eigenvector $\mathbf{u}$ of B_{odd}. As before, we can construct an eigenvector $\mathbf{v}$ of B': for $n_f + n_r + 1 \leq i \leq n$ let $v_i = u_i$; let $v_i = 0$ otherwise. We again use the matrix T to transform $\mathbf{v}$ into an eigenvector $\mathbf{w}$ of B:

$$\mathbf{w} = T\mathbf{v} = \begin{bmatrix} I_{n_f} & 0 & 0 \\ 0 & \frac{1}{\sqrt(2)} I_{n_r} & \frac{1}{\sqrt(2)} I_{n_r} \\ 0 & \frac{1}{\sqrt(2)} I_{n_r} & \frac{-1}{\sqrt(2)} I_{n_r} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \mathbf{v}_{\phi(r)} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{\sqrt(2)} \mathbf{v}_{\phi(r)} \\ \frac{-1}{\sqrt(2)} \mathbf{v}_{\phi(r)} \end{bmatrix}$$

This is an odd eigenvector of B. Since $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$ all have the same eigenvector λ, the claim about eigenvalues of B_{odd} corresponding to odd eigenvalues of B holds. It is easy to show that if two eigenvectors $\mathbf{u_1}$ and $\mathbf{u_2}$ of B_{odd} are orthogonal, then the corresponding eigenvectors $\mathbf{w_1}$ and $\mathbf{w_2}$ are also orthogonal. Thus of an eigenvalue of B_{odd} has multiplicity 2, there are two orthogonal odd eigenvectors of B with that eigenvalue.

Note that if we transform all eigenvectors of B_{even} and B_{odd} this way we get n orthogonal eigenvectors of B (i.e., a full set).

It is easy to construct the components G_{odd} and G_{even} from G. We now give the rules for these constructions; they are easy to verify from the matrix entries of B'.

G_{odd} is a weighted graph on V_r. Start with the subgraph of G induced by V_r, with the weight of vertex v_i equal to the weight of v_i in G and the weight of edge (v_i, v_j) equal to its weight in G. Adjust the weights according to the following two rules:

- **Degree weight adjustment rule:** For each vertex $v_i \in V_r$, if there is an edge in G from v_i to $v_{\phi(i)}$ (i.e., there is an edge from v_i to its image under the automorphism), then increase the weight of v_i in G_{odd} by $w_{i\phi(i)}$.

- **Edge weight adjustment rule:** For each pair of distinct vertices $v_i,\ v_j \in V_r$ and $i < j$, if there is an edge $(v_i, v_{\phi(j)})$ in G (i.e., if there is an edge from v_i in the representative set to the image of v_j outside of the representative set), add an edge (v_i, v_j) with weight $-w_{i,\phi(j)}$ to G_{odd} (if there is already an edge (v_i, v_j) in G_{odd}, subtract $w_{i,\phi(j)}$ from its weight). To see that this rule is well-formed with respect to permutations of the vertex numberings, note that since ϕ is an automorphism of order 2, the existence of an edge $(v_i, v_{\phi(j)})$ in G implies the existence of an edge $(v_{\phi(i)}, v_j)$ in $E(G)$ with the same weight. That is, for any $i < j$ we have

$$w_{ij}^{G_{odd}} = w_{ij}^{G} - w_{i\phi(j)}^{G} = w_{ji}^{G} - w_{j\phi(i)}^{G},$$

where the superscripts on the weights indicate the graph for which they're defined. Thus, if we renumbered the vertices in a way that assigns v_j an index less than the new index of v_i, the weight of the corresponding edge in G_{odd} would be unaffected.

Delete any zero-weight edges from G_{odd}.

G_{even} is a weighted graph on $V_r \cup V_f$. Start with the subgraph of G induced by $V_r \cup V_f$, with the weight of vertex $v_i \in V_r \cup V_f$ equal to its degree in G and the weight of each edge (v_i, v_j) equal to its weight in G. Adjust the weights according to the following three rules:

- **Degree weight adjustment rule:** For each vertex $v_i \in V_r$, if there is an edge in G from v_i to $v_{\phi(i)}$ (i.e., there is an edge from v_i to its image under the automorphism), then decrease the weight of v_i in G_{even} by $w_{i\phi(i)}$.

- **$\mathbf{V_r}$-to-$\mathbf{V_f}$ Edge weight adjustment rule:** For each edge (v_i, v_j) in G_{even} such that $v_i \in V_r$ and $v_j \in V_f$ (i.e., for each edge between the fixed vertices and the representative vertices), multiply its weight by $\sqrt{2}$.

- **$\mathbf{V_r}$-to-$\mathbf{V_r}$ Edge weight adjustment rule:** For each pair of distinct vertices $v_i,\ v_j \in V_r$ and $i < j$, if there is an edge $(v_i, v_{\phi(j)})$ in G (i.e., if there is an edge from v_i in the representative set to the image of v_j outside of the representative set), add an edge (v_i, v_j) with weight $w_{i,\phi(j)}$ to G_{even} (if there is already an edge (v_i, v_j) in G_{even}, add $w_{i,\phi(j)}$ to its weight). We leave it to the reader to check that this rule is well-formed with respect to permutations of the vertex numberings; the argument is the same as for the edge weight adjustment rule in the odd case.

Delete any zero-weight edges from G_{even}.

Since

$$|V(G_{odd})| + |V(G_{even})| = |V_f| + 2|V_r| = |V_f| + |V_m| = |V(G)|,$$

the claim about the combined size of the two graphs holds.

□

It is useful to consider an example of even-odd graph decomposition before proceeding. We will demonstrate one way in which a complete binary tree of three levels can be decomposed. The initial graph is shown in Figure 3.1 below. The first automorphism we use maps leaves with the same parent to each other. Applying the rules for constructing the odd and even components given at the end of the proof above, we get the result shown in Figure 3.1. The odd component is fully decomposed; we can apply one more step of decomposition to the even component using the automorphism that maps the corresponding vertices at the lowest two levels to each other. The

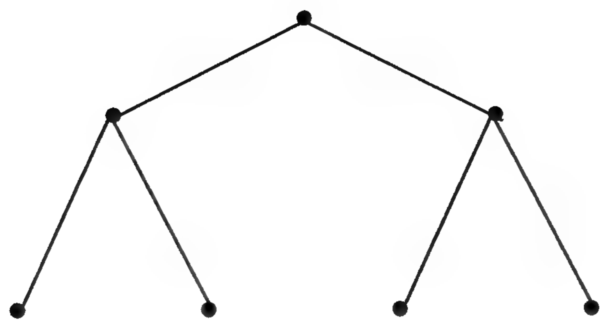

Figure 3: The Initial Graph

result is shown in Figure 3.1. This example will also be useful in arguments about bounding the smallest nonzero eigenvalue of complete binary trees below.

The following technical lemmas about the eigenvalues and eigenvectors of weighted path graphs will be useful in subsequent results.

Lemma 3.5 [Zero Entries of Path Graph Eigenvectors Lemma] *Let B be the standard matrix representation of a weighted path graph G on n vertices. For any vector $\mathbf{x}$ such that $B\mathbf{x} = \lambda\mathbf{x}$ for some real λ, $x_n = 0$ implies $\mathbf{x} = 0$. Likewise, $x_1 = 0$ implies $\mathbf{x} = 0$. If there are two consecutive elements x_i and x_{i+1} that are both zero, then $\mathbf{x} = 0$.*

Proof: We will prove the first result by induction. The base case is for a 2×2 matrix with diagonal entries b_{11} and b_{22}, and off-diagonal entries $b_{12} = b_{21} = -c$. Let $\mathbf{x}$ and λ be as specified by the lemma statement, and assume that $x_2 = 0$. The second element of the vector resulting from multiplying $B\mathbf{x} = \lambda\mathbf{x}$ is $-c \cdot x_1 = \lambda x_2 = 0$. Since $c \neq 0$ by definition (G is a weighted path graph), we must have $x_1 = 0$, which implies that $\mathbf{x} = 0$.

For the induction step, assume that the result holds for all $i \leq k$, and consider the standard matrix representation of a weighted path graph on $k + 1$ vertices. Let the weight of the edge from v_k to v_{k+1} be c. Let $\mathbf{x}$ and λ be as stated, and assume that $x_{k+1} = 0$. For the $k + 1^{\text{st}}$ entry of $B\mathbf{x}$ we have $-c \cdot x_k = \lambda x_{k+1} = 0$. Thus $x_k = 0$. Let $\mathbf{x}'$ be the subvector of $\mathbf{x}$ consisting of the first k entries. Note that with $x_{k+1} = 0$ we have that $\mathbf{x}'$ and λ meet the lemma conditions for the principle leading minor B_k of B, and that $x'_k = 0$. But the leading principle minor B_k is the standard matrix representation for the weighted path graph derived from G by deleting the last edge and vertex. Thus, by the induction hypothesis $\mathbf{x}'$ must be 0; because $x_{k+1} = 0$ this implies that $\mathbf{x} = 0$

A symmetric argument implies the result for $x_1 = 0$. Details are left to the reader.

Again let B be the standard matrix representation of a weighted path graph G. Let $\mathbf{x}$ be a vector meeting the lemma conditions for λ, and assume that $\mathbf{x}$ has two consecutive zero elements x_i and x_{i+1}. If either $i = 1$ or $i + 1 = n$, $\mathbf{x} = 0$ by the previous argument. Otherwise, $x_{i+1} = 0$ implies that the first i elements of $\mathbf{x}$ and λ meet the lemma conditions for the leading principle minor B_i of B. Note that B_i is the standard matrix representation for some weighted path graph. Thus by our previous result the first i entries of $\mathbf{x}$ are zero. By a symmetric argument for the trailing principle minor, the last $n - i$ entries must also be zero, which gives $\mathbf{x} = 0$.

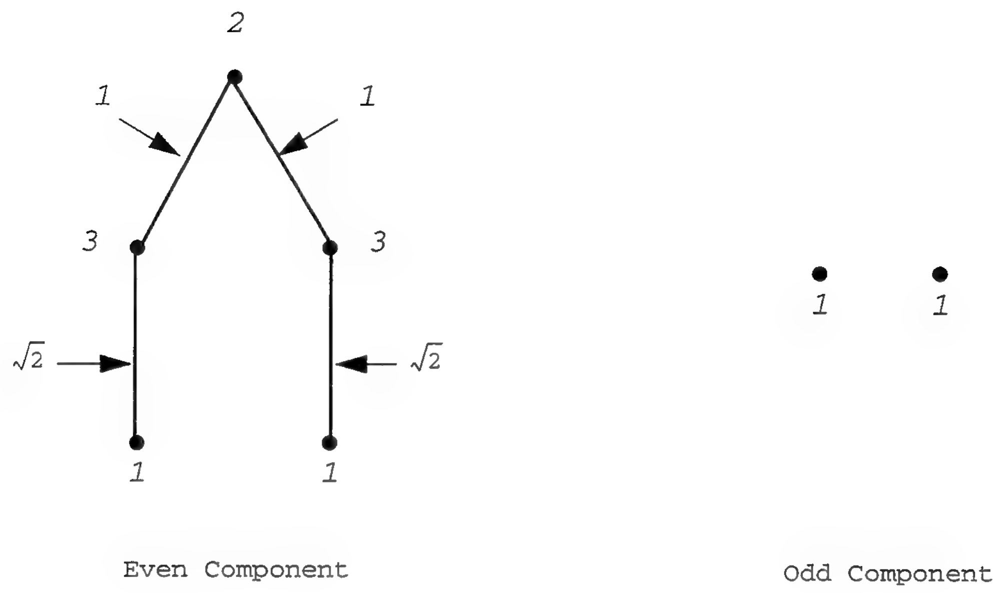

Figure 4: After the First Decomposition Step

□

Since eigenvectors are by definition not equal to the zero vector, the theorem above implies that for eigenvectors of the standard matrix representation B of any weighted path graph, neither the first nor the last entry is zero. Likewise, such an eigenvector cannot have two consecutive zero entries. We can use this to give a simple proof of the following lemma (for a different proof, see e.g. pp. 910-911 of [YG73]).

Lemma 3.6 *All eigenvalues of the standard matrix representation B of a weighted path graph G on n vertices are simple (i.e., have multiplicity one).*

Proof: Let $\mathbf{u}$ and $\mathbf{u}'$ be any two eigenvectors of B for the eigenvalue λ. By the Zero Entries of Path Graph Eigenvectors Lemma (Lemma 3.5), we have $u_n \neq 0$ and $u'_n \neq 0$. Let α be u'_n/u_n; α is non-zero and real. Then $B(\alpha\mathbf{u} - \mathbf{u}') = \lambda(\alpha\mathbf{u} - \mathbf{u}')$. But the n^{th} element of $(\alpha\mathbf{u} - \mathbf{u}')$ is 0, so by Lemma 3.5, we must have $\alpha\mathbf{u} = \mathbf{u}'$, so $\mathbf{u}$ must be a scalar multiple of $\mathbf{u}'$; it is not a distinct eigenvector.

□

A path graph on n vertices has exactly one automorphism of order two: $\phi(i) = n - i + 1$. Thus we can talk about odd and even eigenvectors of a path graph without ambiguity; they are always with respect to this automorphism.

Lemma 3.7 *The eigenvector $\mathbf{u}_2$ corresponding to the second smallest eigenvalue λ_2 of the Laplacian B of a path graph G on n vertices is odd.*

Proof: By Lemma 3.6, we know that $\mathbf{u}_2$ is simple, so by Corollary 3.3, $\mathbf{u}_2$ must be either even or odd. Assume that it is even. We will show that this leads to a contradiction.

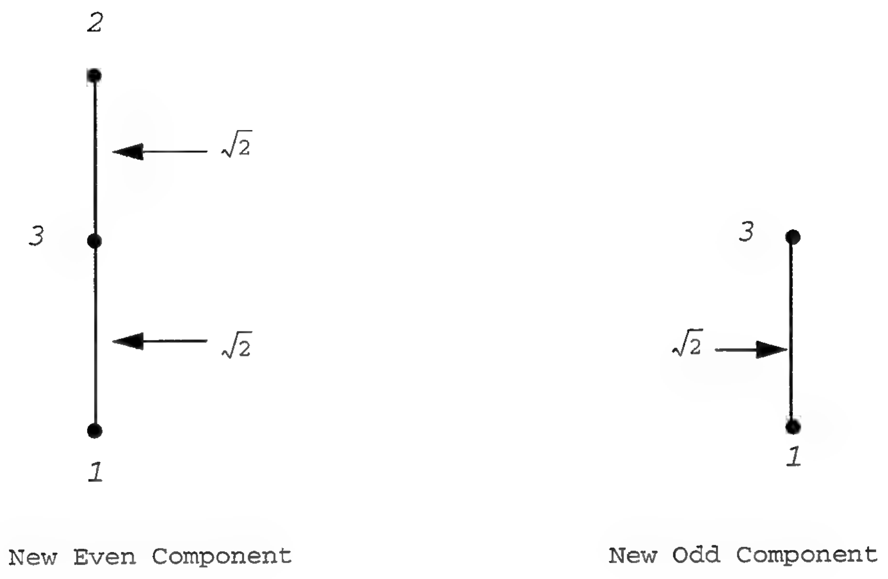

Figure 5: Result of Decomposing Odd Component

Recall the characterization

$$\lambda_2 = \min_{\mathbf{x}\perp\vec{1}} \frac{\mathbf{x}^T A\mathbf{x}}{\mathbf{x}^T\mathbf{x}}.$$

There are two cases we must keep track of: n is odd, and n is even. If n is odd, there is a single center vertex $v_{\lceil \frac{n}{2}\rceil}$ (we will index the vertices along the path from 1 to n). If n is even, there are two center vertices with indices $\frac{n}{2}$ and $\frac{n}{2}+1$; since we have assumed $\mathbf{u}_2$ is even, their entries in $\mathbf{u}_2$ are equal. Thus, by Lemma 3.5, if n is even the eigenvector entries corresponding to the center vertices are non-zero. If n is odd, $\mathbf{u}_2$ is even, and the eigenvector entry for the center vertex is 0, it is easy to check that changing the signs of all eigenvector entries with index less than the center index gives an odd eigenvector with eigenvalue λ_2, which contradicts the simplicity of λ_2. Thus, our assumption that $\mathbf{u}_2$ is even implies that the eigenvector entries corresponding to the center vertex or vertices must be non-zero. Let this value be c.

Now consider the vector $\mathbf{x} = (-c)\cdot\vec{1} + \mathbf{u}_2$. Recall that $\mathbf{u}_2$ is orthogonal to $\vec{1}$. It is easy to see that $\mathbf{x}$ is even, and since $c \neq 0$, we have that

$$\frac{\mathbf{x}^T A\mathbf{x}}{\mathbf{x}^T\mathbf{x}} = \frac{c^2\cdot\left(\vec{1}^T A\vec{1}\right) + \mathbf{u}_2^T A\mathbf{u}_2}{\left((-c)\cdot\vec{1}+\mathbf{u}_2\right)^T\left((-c)\cdot\vec{1}+\mathbf{u}_2\right)} = \frac{\mathbf{u}_2^T A\mathbf{u}_2}{c^2 n + \mathbf{u}_2^T\mathbf{u}_2} < \frac{\mathbf{u}_2^T A\mathbf{u}_2}{\mathbf{u}_2^T\mathbf{u}_2}.$$

However, the entries of $\mathbf{x}$ corresponding to the center vertex or vertices are 0, so as we did above, we can create an odd vector $\mathbf{y}$ such that

$$\frac{\mathbf{y}^T A\mathbf{y}}{\mathbf{y}^T\mathbf{y}} = \frac{\mathbf{x}^T A\mathbf{x}}{\mathbf{x}^T\mathbf{x}}$$

as follows: set $y_i = x_i,\ i < \frac{n}{2}$ and $y_i = -x_i,\ i > \frac{n}{2}$. $\mathbf{y}$ is orthogonal to $\vec{1}$, so it meets the criteria for the characterization of λ_2, so our assumption that $\mathbf{u}_2$ is even gives $\lambda_2 < \lambda_2$, a contradiction. □

3.2 Bounds on λ_2 for Trees and Double Trees

We now use the results from the preceding section to prove some upper and lower bounds on λ_2 for the Laplacian of the complete binary tree and for the Laplacian of a graph we refer to as the **double tree**. We define a double tree as two complete binary trees of k levels for some $k > 0$ connected by an edge between their respective roots. The bounds developed below will be useful in Section 5.

To bound the size of λ_2 of a complete binary tree, we first apply the Even-Odd Decomposition Theorem (Theorem 3.4) a number of times to show that the eigenvalues of a balanced binary tree can be computed from a few simple types of weighted graphs. We then bound the eigenvalues for these types of graphs using the interlacing theorems and matrix perturbation techniques. We show that for $k \geq 3$, a complete binary tree on k levels with $n = 2^k - 1$ vertices, $\lambda_2 = \Theta(\frac{1}{n})$. More specifically, $\frac{1}{n} < \lambda_2 < \frac{2}{n}$.

For the following argument, we will assume that we are working with a complete binary tree of $k > 2$ levels. As above, $n = 2^k - 1$.

We start by applying the Even-Odd Graph Decomposition Theorem with respect to the automorphisms that each map one pair of neighboring leaves to each other. There are 2^{k-1} leaves and 2^{k-2} such automorphisms. For the odd eigenvalues, we get $\lambda = 1$ for each of the 2^{k-2} single-vertex graphs that comprise the odd components. The result for $k = 3$ was shown above in Figure 3.1.

To help in understanding the structure of the even component that results from the preceding sequence of decompositions, we introduce the following terminology: for $i \geq 3$, the i^{th} **odd collapsed tree graph** is a weighted graph on $i - 1$ vertices, with every edge having weight $\sqrt{2}$, vertex v_1 having weight 1, and all other vertices having weight 3 (the odd collapsed tree graph for $i = 5$ is shown in Figure 3.2 below). With this terminology, we can describe the resulting G_{even} in the following way: the even component has the structure of a complete binary tree of $k - 1$ levels, except that the leaves of the tree are replaced by the odd collapsed tree graph for $i = 3$; the vertices at level $k - 2$ are connected to the pseudo-leaves by edges of weight 1 to the vertices v_2 of the odd collapsed tree graphs. This is illustrated for $k = 3$ in Figure 3.1 above.

We now repeat the process for the 2^{k-3} automorphisms that map neighboring pairs of the odd collapsed tree graphs to each other. The odd eigenvalues found during these steps are the eigenvalues of the odd collapsed tree graph for $i = 3$, each occurring 2^{k-3} times, plus the eigenvalues of the new even component, a weighted graph that looks like a complete binary tree with $k - 2$ levels, except that the leaves are replaced by odd collapsed tree graphs for $i = 4$. Once again, each pseudo-leaf is connected to its parent by an edge of weight 1 from vertex v_3 of the odd collapsed tree graph.

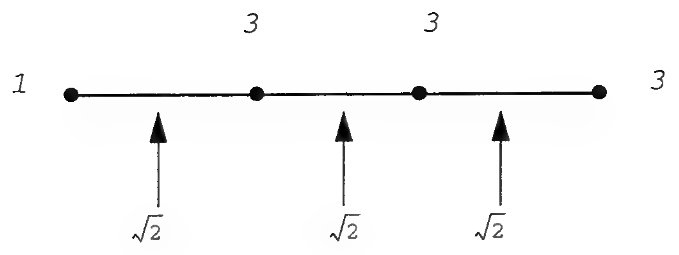

Figure 6: Odd Collapsed Tree Graph, $i = 5$

We continue this process; at the j^{th} series of reductions we get eigenvalues for the odd collapsed tree graph for $i = j + 1$ and an even component that is a weighted graph that looks like a complete binary tree with $k - j$ levels, except that the leaves are replaced by odd collapsed tree graphs for $i = j + 2$.

At the last decomposition step, we start with a weighted graph that looks like a complete binary tree with 2 levels, except that the leaves have been replaced by odd collapsed tree graphs for $i = k$. Applying the Even-Odd Decomposition Theorem one last time, we find that the eigenvalues for this graph are those of the odd collapsed tree graph for $i = k$ plus the eigenvalues for a graph on k vertices that consists of the odd collapsed tree graph for $i = k$ plus vertex v_k with weight 2, and edge (v_{k-1}, v_k) with weight $\sqrt{2}$ (we will refer to this latter graph as the **even collapsed tree graph** for $i = k$; the even collapsed tree graph for $i = 5$ is shown in Figure 3.2 below). Recall that the

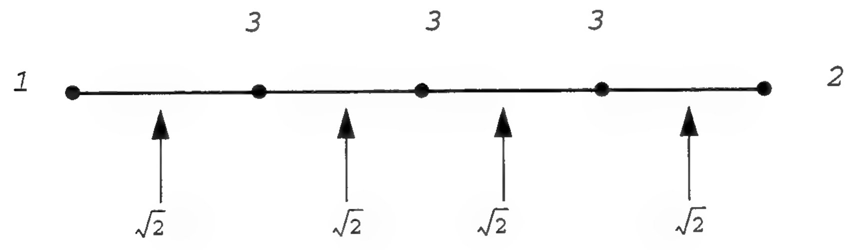

Figure 7: Even Collapsed Tree Graph, $i = 5$

eigenvalues not represented by either of these final components are either 1 (from the first set of decompositions) or eigenvalues for the odd collapsed tree graphs for i, $3 \leq i < k$.

Let $\mu^{(k)}$ be the smallest eigenvalue of the odd collapsed tree graph for $i = k$. We claim that λ_2 for the complete binary tree on k levels is equal to $\mu^{(k)}$. $\mu^{(k)}$ is less than $\mu^{(i)}$ for any $i < k$ by the First Interlacing Theorem. We will show in the argument below that $\mu^{(k)} < 1$ for $k \geq 3$. It is easy to show that the standard matrix representation for the even collapsed tree graph for $i = k$ is singular, so its smallest eigenvalue is 0. Thus, an application of the First Interlacing Theorem gives us that the smallest eigenvalue of the odd collapsed tree graph for $i = k$ is less than or equal to the first non-zero eigenvalue of the even collapsed tree graph for $i = k$ (the standard matrix representation of the former is a leading principle submatrix of the latter).

Bounds on $\mu^{(k)}$ can be computed using perturbation techniques. Note that the standard matrix representation for the odd collapsed tree is nonsingular, so the smallest eigenvalue is the one of interest.

For the rest of this argument, we will use two matrices. The first, B, is the standard matrix representation for the odd collapsed tree graph for $i = k$. The determinant of B is 1 (this is an easy induction proof on k). The vertex ordering is as given in the description of the graph given above. The second matrix, B', has

$$b'_{11} = b_{11} - \frac{1}{2^{k-1}-1} = 1 - \frac{1}{2^{k-1}-1} = \frac{n-3}{n-1},$$

and all other entries are the same as for B. The determinant of B' is 0 (again, easily shown via an easy induction proof).

Let $\mathbf{v}$ be the eigenvector of B' corresponding to $\lambda = 0$ (we'll assume that it's scaled to unit length). Then we have

$$\mathbf{v}^T B' \mathbf{v} = \mathbf{v}^T B \mathbf{v} - \frac{v_1^2}{2^{k-1}-1} = 0.$$

By the Courant-Fischer Minimax Theorem (see, e.g., [GL89]), $\frac{v_1^2}{2^{k-1}-1}$ is an upper bound on $\mu^{(k)}$ for B. We will now show that for $k \geq 3$, this upper bound is strictly less than $\frac{2}{n}$. Note that this will immediately imply that $\mu^{(k)}$ is less than 1.

Because $2^{k-1} - 1 = \frac{n-1}{2}$, we will have that $\frac{v_1^2}{2^{k-1}-1} < \frac{2}{n}$ if $v_1^2 < \frac{n-1}{n}$. Assume that this is not so; we will show that this assumption contradicts the fact that $\mathbf{v}$ is unit length. Since $B'\mathbf{v} = 0$, the definition of B' gives us that

$$b_{11}v_1 - \sqrt{2}v_2 = \frac{n-3}{n-1}v_1 - \sqrt{2}v_2 = 0 \rightarrow v_2 = \frac{1}{\sqrt{2}}\left(\frac{n-3}{n-1}\right)v_1,$$

and by our assumption

$$v_2^2 = \frac{1}{2}\left(\frac{n-3}{n-1}\right)^2 v_1^2 \geq \frac{1}{2n}\frac{(n-3)^2}{n-1}.$$

Since $\mathbf{v}$ is unit length, our assumption yields

$$1 = \mathbf{v}^T\mathbf{v} = \sum_{i=1}^{k-1} v_i^2 \geq v_1^2 + v_2^2 = \frac{1}{n}\left(n - 1 + \frac{1}{2}\frac{(n-3)^2}{n-1}\right) > 1,$$

a contradiction (the last inequality holds because for $k \geq 3$, we have $n \geq 7$, and thus $\frac{(n-3)^2}{n-1} > 2$).

To get a lower bound on $\mu^{(k)}$, let $\mathbf{u}$ be the unit-length vector such that $\mathbf{u}^T B\mathbf{u}$ is minimized (i.e., $\mathbf{u}^T B\mathbf{u} = \mu^{(k)}$). We have:

$$\mathbf{u}^T B'\mathbf{u} = \mathbf{u}^T B\mathbf{u} - \frac{u_1^2}{2^{k-1}-1} = \mu^{(k)} - \frac{u_1^2}{2^{k-1}-1} \geq 0.$$

The last inequality holds by a version of the Second Interlacing Property with $\alpha < 0$; B' has a zero eigenvalue, and all its other eigenvalues will be greater than or equal to the (positive) eigenvalues of B. B' is thus positive semidefinite.

The above equation implies that if we can show that $u_1 > \frac{1}{\sqrt{2}}$, we will have $\mu^{(k)} > \frac{1}{2^k-2} > \frac{1}{n}$. Since B is tridiagonal, this is easily done by using B and $\mu^{(k)}$ to generate a recurrence for the entries of $\mathbf{u}$.

Consider the first entry of $B\mathbf{u}$. We have $u_1 - \sqrt{2}u_2 = \mu^{(k)}u_1$. Since $1 > \mu^{(k)} > 0$, we have $u_2 < \frac{u_1}{\sqrt{2}}$. This provides the base case for a proof by induction that $u_{i+1} < \frac{u_i}{\sqrt{2}}$. The details are left to the reader; the previous inequality serves as the induction hypothesis; the matrix B gives us

$$3u_i - \sqrt{2}(u_{i-1} + u_{i+1}) = \mu^{(k)}u_i,$$

which can be used to prove the desired result for u_{i+1}.

Since $\mathbf{u}$ is unit length, we have

$$1 = \mathbf{u}^T\mathbf{u} = \sum_{i=1}^{k-1} u_i^2 < \sum_{i=1}^{k-1}\left(\frac{1}{\sqrt{2}}\right)^{2(i-1)} u_1^2 = u_1^2\sum_{i=1}^{k-1}\left(\frac{1}{2}\right)^{(i-1)} < 2u_1^2,$$

which gives the desired result that $u_1 > \frac{1}{\sqrt{2}}$.

Thus we have proved the following lemma:

Lemma 3.8 *For a complete balanced binary tree on $k \geq 3$ levels and $n = 2^k - 1$ vertices, we have $\frac{1}{n} < \lambda_2 < \frac{2}{n}$.*

For double trees where each of the component trees has k levels and $n = 2^{k+1} - 2$ vertices, we have the following lemma:

Lemma 3.9 *For a double tree on $n \geq 14$ vertices, we have $\frac{1}{n} < \lambda_2 < \frac{4}{n}$.*

Proof: The proof will make use of a number of facts from the proof of the λ_2 bounds for complete binary trees given above.

We start by noting that we can apply the Even-Odd Decomposition Theorem starting from the leaves of the trees as we did above. The odd eigenvalues determined at each step are the same as for the complete binary tree.

At the last decomposition step (the one that divides between the two roots) results in two components: the even component G_e, which is the same as even component for a tree of k levels, and the odd component G_o, which is like the $k+1$st odd collapsed tree graph, except that the weight of vertex k is 4 rather than 3. Again, the standard matrix representation is singular for G_e and non-singular for G_o; an application of the Second Interlacing Theorem thus implies that λ_2 for the double tree is the smallest eigenvalue for G_o.

Next, note that the standard matrix representation of the odd collapsed tree for $i = k$ is a principle submatrix of the standard matrix representation for G_o. By the First Interlacing Theorem and the analysis of the odd collapsed tree above, we immediately have that $\lambda_2 < \frac{2}{2^k-1}$; since the double tree has $n = 2^{k+1} - 2$ vertices, this implies that $\lambda_2 < \frac{4}{n}$.

Finally, note that G_o is the odd collapsed tree for $i = k+1$ with 1 added to the weight of vertex k. We can thus apply the Second Interlacing Theorem and the results for the odd collapsed tree above (recall that in the proof we showed that $\mu^{(k)} > \frac{1}{2^k-2}$) to show that $\lambda_2 \geq \mu^{(k+1)} > \frac{1}{2^{k+1}-2}$; since the double tree has $n = 2^{k+1} - 2$ vertices, this implies that $\lambda_2 > \frac{1}{n}$.

This completes the proof of the lemma.

□

4 A Bad Family of Bounded-Degree Planar Graphs for Spectral Bisection

In this section we present a family of bounded-degree planar graphs that have constant-size separators. However, the separators produced by spectral bisection have size $\Theta(n)$ for both edge and vertex separators. Since there are algorithms that produce $O(\sqrt{n})$ vertex separators for planar graphs, and hence $O(\sqrt{n})$ edge separators for bounded-degree planar graphs, spectral bisection performs poorly on these graphs relative to other algorithms.

The family of graphs is parameterized on the positive integers. G_k consists of two path graphs, each on $3k$ vertices, with a set of edges between the two paths as follows: label the vertices of one path from 1 to $3k$ in order (the **upper path**), and label the other path from $3k+1$ to $6k$ in order (the **lower path**). For $1 \leq i \leq k$ there is an edge between vertices $2k+i$ and $5k+i$. This was shown in Figure 1 in the Introduction. It is obvious that G_k is planar for any k, and that the maximum degree of any vertex is 3.

Note that the graph has the approximate shape of a cockroach, with the section containing edges between the upper and lower paths being the body, and the other sections of the paths being antennae. This terminology will allow us to refer easily to parts of the graph.

G_k has one automorphism of order 2 that maps the vertices of the upper path to the vertices of the lower path and vice versa. For the rest of this section, the terms "odd vector" and "even vector" will be used with respect to this automorphism. Thus, an even vector $\mathbf{x}$ has $x_i = x_{3k+i}$ for all i in the range $1 \leq i \leq 3k$; an odd vector $\mathbf{y}$ has $y_i = -y_{3k+i}$ for all i, $1 \leq i \leq 3k$.

We can now discuss the structure of the eigenvectors of the Laplacian of G_k.

Lemma 4.1 *Any eigenvector $\mathbf{u}_i$ with eigenvalue λ_i of the Laplacian B_k of G_k can be expressed in terms of linear combinations of even and odd eigenvectors with eigenvalue λ_i. In particular, $\mathbf{u}_i$ can be expressed as a linear combination of:*

- *an even eigenvector of B_k in which the values associated with the upper path are the same as for the eigenvector with eigenvalue λ_i (if it exists) of a path graph on $3k$ vertices, and*
- *an odd eigenvector of B_k in which the values associated with the upper path are the same as for the eigenvector with eigenvalue λ_i (if it exists) of a weighted graph that consists of a path graph on $3k$ vertices for which the vertex weights of v_{2k+1} through v_{3k} have been increased by 2.*

Proof: The first claim follows by the Even-Odd Eigenvector Theorem applied with respect to the automorphism that maps the vertices of the upper path to the vertices of the lower path and vice versa.

The claim about the specific structure of the $\mathbf{u}_i$'s follows by a direct application of the construction used in the proof of the Even-Odd Graph Decomposition Theorem with respect to the same automorphism.

□

We now give the proof that spectral bisection gives bad separators for the family of graphs G_k.

Theorem 4.2 *Spectral bisection produces $\Theta(n)$ edge and vertex separators for G_k for any k.*

Proof: The first step is to show that $\mathbf{u}_2$ is odd. Intuitively, this implies that the spectral method will split the graph into the upper path and the lower path.

Recall that $\lambda_2 = \min_{\mathbf{x} \perp \vec{1}} \frac{\mathbf{x}^T B_k \mathbf{x}}{\mathbf{x}^T \mathbf{x}}$. We will construct an odd vector $\mathbf{x}$ such that the quotient $\frac{\mathbf{x}^T B_k \mathbf{x}}{\mathbf{x}^T \mathbf{x}}$ is less than $\frac{\mathbf{y}^T B_k \mathbf{y}}{\mathbf{y}^T \mathbf{y}}$ for any even eigenvector $\mathbf{y}$ orthogonal to $\vec{1}$ ($\vec{1}$ is the smallest even eigenvector). To do this, we need to show that $\frac{\mathbf{x}^T B_k \mathbf{x}}{\mathbf{x}^T \mathbf{x}}$ is less than the second smallest even eigenvalue. From Lemma 4.1 above, we know that the second smallest even eigenvalue of B_k is the same as the second smallest eigenvalue of the Laplacian of a path graph on $3k$ vertices; it is well-known that this value is $4\sin^2(\frac{\pi}{6k})$ (see for example [Moh88]).

Let $\mathbf{z}$ be the eigenvector corresponding to the second smallest eigenvalue μ_2 for the Laplacian B of the path graph G on $4k$ vertices ($\mu_2 = 4\sin^2(\frac{\pi}{8k})$). Construct $\mathbf{x}$ as follows:

$$x_i = \begin{cases} z_i & 1 \le i \le 2k, \\ z_{7k-i+1} & 3k+1 \le i \le 5k, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

That is, we assign the first $2k$ values from the path G to the upper antenna of the roach, working in the direction towards the body, and we assign the last $2k$ entries from G to the lower antenna, working from the body outward. Since $\mathbf{z}$ and $\mathbf{x}$ have the same set of non-zero entries, $\mathbf{x}^T\mathbf{x} = \mathbf{z}^T\mathbf{z}$. Likewise, since $\mathbf{z}$ is orthogonal to the "all-ones" vector, so is $\mathbf{x}$.

In order to see that $\mathbf{x}^T B_k \mathbf{x} < \mathbf{z}^T B \mathbf{z}$, recall (1) from Section 3: for Laplacian A and vector $\mathbf{y}$ we have

$$\mathbf{y}^T A \mathbf{y} = \sum_{(v_i,v_j)\in E} (y_i - y_j)^2.$$

For every edge in G except one, there is an edge in G_k that contributes the same value to this sum. The one exception is the edge (v_{2k}, v_{2k+1}). Since $\mathbf{z}$ is an odd vector by Lemma 3.7, and since $\mathbf{z}$ has an even number of entries, $z_{2k} = -z_{2k+1}$. By the Zero Entries of Path Graph Eigenvectors Lemma (Lemma 3.5), it is not possible for both z_{2k} and z_{2k+1} to be zero, so z_{2k} is equal to some non-zero value c, and this edge contributes $4c^2$ to the value of $\mathbf{z}^T B \mathbf{z}$. On the other hand, there are two edges in G_k that contribute non-zero values and that do not have corresponding edges in G: (v_{2k}, v_{2k+1}) and (v_{5k}, v_{5k+1}). Each of these edges contributes c^2 to $\mathbf{x}^T B_k \mathbf{x}$. Thus we have

$$\mathbf{x}^T B_k \mathbf{x} = \mathbf{z}^T B \mathbf{z} - 4c^2 + 2c^2 = \mathbf{z}^T B \mathbf{z} - 2c^2 < \mathbf{z}^T B \mathbf{z}.$$

Since $\mathbf{x}^T\mathbf{x} = \mathbf{z}^T\mathbf{z}$, this gives us

$$\lambda_2(G_k) \leq \frac{\mathbf{x}^T B_k \mathbf{x}}{\mathbf{x}^T\mathbf{x}} < \frac{\mathbf{z}^T B\mathbf{z}}{\mathbf{z}^T\mathbf{z}} = 4\sin^2(\frac{\pi}{8k}) < 4\sin^2(\frac{\pi}{6k}).$$

That is, the second smallest eigenvalue of B_k is less than any non-zero even eigenvalue, and is thus odd by the Even-Odd Eigenvector Theorem (Theorem 3.2).

There are a few details to finish off. In particular, we need to show that there aren't too many zero entries in $\mathbf{u}_2$ (spectral bisection as defined in this paper won't separate vertices with the same value). Since $\mathbf{u}_2$ is an odd vector and since the odd component of G_k is a weighted path graph, Lemmas 3.5 (the Zero Entries of Path Graph Eigenvectors Lemma) and 4.1 imply that $\mathbf{u}_2$ cannot have consecutive zeros, and the values corresponding to vertices $3k$ and $6k$ are non-zero. Thus the edge separator generated by spectral bisection must cut at least half the edges between the upper and lower paths; since none of these edges share an endpoint, the cover used in generating the vertex separator must include at least this number of vertices. The theorem follows.
□

5 A Bad Family of Graphs for the "Best Threshold Cut" Algorithm

While the roach graph defeats spectral bisection, the second smallest eigenvector can still be used to find a small separator using the "best threshold cut" algorithm. In particular, Theorem 3.1 implies that considering all threshold cuts induced by $\mathbf{u}_2$ will let us find a constant-size cut: If q_{min} is the minimum cut quotient for these cuts, then

$$q_{min} \leq \sqrt{\lambda_2(2\Delta - \lambda_2)} \leq \frac{\sqrt{6}\pi}{4k},$$

which implies q_{min} is $O(\frac{1}{n})$. Since the denominator of q_{min} is less than or equal to $\frac{n}{2}$, the number of edges in this cut must be bounded by a constant.

In this section we show that there is a family of graphs for which the "best threshold cut" algorithm does poorly. Recall that in Section 3.2 we defined a double tree as a graph that consists of two complete binary trees of k levels for some $k > 0$, connected by an edge between their respective roots. The **tree-cross-path graph** consists of the crossproduct of a double tree on p_1 vertices and a path graph on p_2 vertices. We will show below that there are tree-cross-path graphs that will defeat the "best threshold cut" algorithm. To do so, we will use the bounds from Section 3.2 on λ_2 for trees and double trees.

We can formally state the result for this section as follows:
Theorem 5.1 *There exists a graph G for which the "best threshold cut" algorithm finds a separator S such that the cut quotient for S is bigger than $i(G)$ by a factor as large (to within a constant) as allowed by the theoretical bounds provided by Theorem 3.1.*
Proof: Let G be the tree-cross-path graph that is the crossproduct of a double tree of size p and a path of length $cp^{\frac{1}{2}}$ for some c in the range $3.5 \leq c < 4$. To insure that we have integer sizes for the tree and the path, we restrict p to integers of the form $2^k - 2$ for $k > 2$. Then we choose c in the range specified such that $cp^{\frac{1}{2}}$ is an integer (by our choice of p there will be an integer in this range).

Recall that the eigenvalues of a graph crossproduct are all pairwise sums of the eigenvalues from the graphs used in the crossproduct operation. Let ν_2 be the second smallest eigenvalue of the double tree on p vertices, and let μ_2 be the second smallest eigenvalue for the path on $cp^{\frac{1}{2}}$ vertices. If $\mu_2 < \nu_2$, then λ_2 for the crossproduct will be μ_2 (i.e., μ_2 added to the zero eigenvalue of the double tree). But we have that $\mu_2 = 4\sin^2(\frac{\pi}{2cp^{\frac{1}{2}}})$, and that ν_2 is greater than or equal to $\frac{1}{p}$. Therefore we need to show that

$$4\sin^2\left(\frac{\pi}{2cp^{\frac{1}{2}}}\right) < \frac{1}{p}.$$

Reorganizing, simplifying, and noting that $\sin(\theta) < \theta$ for $0 < \theta \leq \frac{\pi}{2}$, we want

$$\frac{\pi}{2cp^{\frac{1}{2}}} < \frac{1}{2p^{\frac{1}{2}}}, \quad \text{or } \pi < c.$$

Clearly by our choice of c this inequality holds.

Since path graph eigenvalues are simple (Lemma 3.6), the second smallest eigenvalue of G will also be simple.

We note that the tree-cross-path graph can be thought of as $cp^{\frac{1}{2}}$ copies of the double tree, each corresponding to one vertex of the path graph. Each vertex in the i^{th} copy of the double tree is connected by an edge to the corresponding vertex in copies $i-1$ and $i+1$. This description allows us to construct the eigenvector for the second smallest tree-cross-path eigenvalue as follows: Assign each vertex in double tree copy i the value for vertex i in the path graph eigenvector for μ_2. Note that this is the only possible eigenvector since the eigenvalue is simple.

Now consider any copy of the double tree: every vertex in that copy gets the same value in the characteristic valuation. Thus the cut S made by the "best threshold cut" algorithm must separate at least two copies of the double tree, and thus must cut at least p edges. There is a bisection S^* of size $cp^{\frac{1}{2}}$ (cut the edge between the roots in each double tree); because this cut is a bisection, the ratio between the cut quotient q for S and $i(G)$ is at least as large as the ratio between the sizes of these cuts:

$$\frac{q}{i(G)} \geq \frac{|S|}{|S^*|} \geq \frac{p}{cp^{\frac{1}{2}}} = \Omega\left(p^{\frac{1}{2}}\right).$$

From Theorem 3.1, we have that

$$\frac{\lambda_2}{2} \leq i(G) \leq q \leq \sqrt{\lambda_2(2\Delta - \lambda_2)}.$$

This plus the fact that the tree-cross-path graph has bounded degree ($\Delta = 5$) implies we must have

$$\frac{q}{i(G)} \leq \frac{2\sqrt{\lambda_2(2\Delta - \lambda_2)}}{\lambda_2} = O\left(\frac{1}{\sqrt{\lambda_2}}\right) = O\left(p^{\frac{1}{2}}\right).$$

These two bounds imply that, to within a constant factor, the ratio is as large as possible, and the theorem holds.

□

6 A Bad Family of Graphs for Generalized Spectral Algorithms

The results of the previous section can be extended to more general algorithms that use some number k (where k might depend on n) of the eigenvectors corresponding to the k smallest non-zero eigenvalues. In particular, consider algorithms that meet the following restrictions:

- The algorithm computes a value for each vertex using only the eigenvector components for that vertex from k eigenvectors corresponding to the smallest non-zero eigenvalues (for convenience, we refer to these as the k **smallest eigenvectors**). The function computed can be arbitrary as long as its output depends only on these inputs.

- The algorithm partitions the graph by choosing some threshold t and then putting all vertices with values greater than t on one side of the partition, and the rest of the vertices on the other side.

- The algorithm is free to compute the break point t in any way; e.g., checking the separator ratio for all possible breaks and choosing the best one is allowed.

We will call such an algorithm **purely spectral**.

The following theorem gives a bound on how well such algorithms do when the number of eigenvectors used is a constant:

Theorem 6.1 *Consider the purely spectral algorithms that use the k smallest eigenvectors for k a fixed constant. Then there exists a family of graphs $\mathcal{G}$ such that $G \in \mathcal{G}$ has a bisection S^* with $|S^*| \geq (k^2 n)^{\frac{1}{3}}$, and such that any purely spectral algorithm using the k smallest eigenvectors will produce a separator S for G such that $|S| \geq \left(\frac{|S^*|}{\pi k+1}\right)^2$.*

Proof: We will show that $\mathcal{G}$ is the set of tree-cross-path graphs that are the crossproducts of double trees of size p (where p is an integer of the form $2^k - 2$ for $k \geq 3$) and paths of length $cp^{\frac{1}{2}}$, where c is a constant chosen such that $\pi k < c \leq \pi k + 1$ and $cp^{\frac{1}{2}}$ is an integer.

Recall that the eigenvalues of a graph crossproduct are all pairwise sums of the eigenvalues from the graphs used in the crossproduct operation. Assume for the moment that the k smallest nonzero eigenvalues of any $G \in \mathcal{G}$ are the same as the k smallest nonzero eigenvalues of the path graph used in defining G. This will clearly hold if we can show that these k eigenvalues are smaller than λ_2 of the double tree; in that case the k smallest non-zero eigenvalues of the crossproduct will be these eigenvalues from the path graph added to the zero eigenvalue of the double tree. Since the path graph eigenvalues are simple (Lemma 3.6), the corresponding tree-cross-path eigenvalues will also be simple.

We note that the tree-cross-path graph can be thought of as $cp^{\frac{1}{2}}$ copies of the double tree, each corresponding to one vertex of the path graph. Each vertex in the ith copy of the double tree is connected by an edge to the corresponding vertex in copies $i-1$ and $i+1$. This description allows us to construct an eigenvector for each of these k tree-cross-path eigenvalues as follows: Assign each vertex in double tree copy i the value for vertex i in the path graph eigenvector for this eigenvalue. Note that these are the only possible eigenvectors since these eigenvalues are simple.

The purely spectral algorithm will produce a cut S with cut quotient q. Recall our assumption about the k smallest eigenvectors and consider any copy of the double tree: since every vertex in that copy gets the same value for each eigenvector, the same value will be assigned to each vertex in this copy by the algorithm. This implies that S must separate at least two copies of the double tree, and thus must cut at least p edges.

There is bisection S^* of size $cp^{\frac{1}{2}}$ (cut the edge between the roots in each double tree); because $n = cp^{\frac{3}{2}}$ and $c > k$ we have $|S^*| > k^{\frac{2}{3}}n^{\frac{1}{3}}$. It is obvious that

$$|S| \geq \left(\frac{|S^*|}{c}\right)^2;$$

since we have $c \leq \pi k + 1$, the claim in the theorem statement holds if our assumption holds.

To prove that the assumption about the form of the k smallest eigenvectors holds for all $G \in \mathcal{G}$, we still need to prove that a path graph on $cp^{\frac{1}{2}}$ vertices has k nonzero eigenvalues smaller than λ_2 for a double tree on p vertices. Recall that the eigenvalues of a path graph on l vertices are $4\sin^2(\frac{\pi i}{2l})$ for $0 \leq i < l$, and that λ_2 for a double tree on p vertices is greater than or equal to $\frac{1}{p}$. Therefore we need to show that

$$4\sin^2\left(\frac{\pi k}{2cp^{\frac{1}{2}}}\right) < \frac{1}{p}.$$

Reorganizing, simplifying, and noting that $\sin(\theta) < \theta$ for $0 < \theta \leq \frac{\pi}{2}$, we have

$$\frac{\pi k}{2cp^{\frac{1}{2}}} < \frac{1}{2p^{\frac{1}{2}}}, \quad \text{or} \quad \pi k < c.$$

Clearly this inequality holds.

□

Note that for the case in which k is constant, we have the following results:

- the cut quotient q_S will be no better than the best cut quotient q_{min} produced by looking at all threshold cuts for $\mathbf{u}_2$, and
- the gap between $i(G)$ and q_{min} is as large as possible (within a constant factor) with respect to Theorem 3.1.

These results can be shown using techniques from the previous section. Thus, G is a graph for which using k eigenvectors does not improve the performance of the "best threshold cut" algorithm.

These results also hold for certain variants of the definition of "purely spectral". For example, Chan, Gilbert, and Teng have proposed using the entries of eigenvectors 2 through $d + 1$ of the Laplacian as spatial coordinates for the corresponding vertices of a graph [CGT94]. The graph is then partitioned using a geometric separator algorithm [MTV91],[GMT95]. If this technique was applied (using a fixed d) to the counterexample graph used in the proof above, all vertices in a particular copy of the double tree would end up with the same coordinates; the geometric algorithm would then have to cut between copies of the double tree, yielding the same bad cuts as in the proof.

6.1 Purely Spectral Algorithms that Use More than a Constant Number of Eigenvectors

There are still a number of open questions about the performance of purely spectral algorithms that use more than a constant number of eigenvectors (in particular, how well can such algorithms do if they are allowed to use all the eigenvectors?). However, just using more than a constant number of eigenvectors is not sufficient to guarantee good separators. In particular, the counterexamples and arguments in the previous sections can be extended to prove the following theorem:

Theorem 6.2 *For large enough n and $0 < \epsilon < \frac{1}{4}$, there exists a bounded-degree graph G on n vertices such that any purely spectral algorithm using the n^ϵ smallest eigenvectors will produce a separator S for G that has a cut quotient greater than $i(G)$ by at least a factor of* $n^{\left(\frac{1}{4}-\epsilon\right)} - 1$.

Proof: Once again, we will take G to be the tree-cross-path graph. As in the previous two proofs, we choose p_1 (the double-tree size) and p_2 (the path size) such that the smallest n^ϵ eigenvalues of the crossproduct are the same as the smallest n^ϵ eigenvalues of the path graph. Once again, a purely spectral algorithm will separate two adjacent double trees, while a better cut would cut the edges between the roots of the double trees. It remains to choose p_1 and p_2 such that the claim about the smallest eigenvalues of the crossproduct holds, and to show that the resulting cut is bad.

We set p_1 to some arbitrary p, subject to the conditions presented below that p be sufficiently large. Then $p_2 = \left\lceil p^{\left(\frac{1}{2}+2\epsilon\right)} \right\rceil$. Note that we can choose p large enough such that

$$p > p^{\left(\frac{1}{2}+2\epsilon\right)} + 1 > p_2.$$

This implies that $p > n^{\frac{1}{2}}$, where $n = p_1 p_2$. Note that this allows us to show easily that $n^\epsilon < p^{2\epsilon} < p_2$ (i.e., that there are sufficiently many eigenvalues from the path graph). We also note that even for fairly small p, $p_2 < 2p^{\left(\frac{1}{2}+2\epsilon\right)}$, which implies that $n < 2p^{\left(\frac{3}{2}+2\epsilon\right)}$.

Now consider the ratio of the cut produced by cutting the double-tree edges versus the cut produced by a purely spectral method under the assumption that the n^ϵ smallest eigenvalues are the same as for the path graph. As in previous proofs, this ratio is at least as large as the ratio between the number of edges cut:

$$\frac{p}{\left\lceil p^{\left(\frac{1}{2}+2\epsilon\right)} \right\rceil} > p^{\left(\frac{1}{2}-2\epsilon\right)} - 1.$$

The inequality will hold for p chosen sufficiently large. Using the fact that p is also big enough that $p > n^{\frac{1}{2}}$, we have

$$p^{\left(\frac{1}{2}-2\epsilon\right)} - 1 > n^{\frac{1}{2}\left(\frac{1}{2}-2\epsilon\right)} - 1 = n^{\left(\frac{1}{4}-\epsilon\right)} - 1.$$

All that is left to prove is the assumption about the smallest eigenvalues. If we let $\alpha = \frac{1}{2} - 2\epsilon$, then $\alpha > 0$ and one of the inequalities above can be written as $n < 2p^{(2-\alpha)}$. Recall that the eigenvalues of a path graph on k vertices are $4\sin^2(\frac{\pi i}{2k})$ for $0 \leq i < k$, and that λ_2 for a double tree on p vertices is greater than or equal to $\frac{1}{p}$. We need to show that for p large enough,

$$4\sin^2\left(\frac{\pi n^\epsilon}{2\left\lceil p^{\left(\frac{1}{2}+2\epsilon\right)}\right\rceil}\right) < \frac{1}{p}.$$

Reorganizing, simplifying, noting that $\sin(\theta) < \theta$ for $0 < \theta \leq \frac{\pi}{2}$, and applying the inequality above, we want to show that there is a p large enough such that

$$\frac{\pi n^\epsilon}{2\left\lceil p^{\left(\frac{1}{2}+2\epsilon\right)}\right\rceil} < \frac{\pi\left(2p^{(2-\alpha)}\right)^\epsilon}{2p^{\left(\frac{1}{2}+2\epsilon\right)}} = \frac{\pi 2^\epsilon p^{-\alpha\epsilon}}{2p^{\frac{1}{2}}} < \frac{1}{2p^{\frac{1}{2}}}, \quad \text{or } \pi 2^\epsilon < p^{\alpha\epsilon}.$$

Clearly this inequality holds for large enough p.
□

6.2 A Final Note About Tree-Cross-Path Graphs

While the tree-cross-path graph appears to be very specialized, we can make the following argument that it has some relation to practice: We noted in Section 3 that the Second Interlacing Theorem implies that adding an edge to a graph G gives a new graph G' with λ_2' greater than or equal to λ_2 for G. Therefore the preceding results holds if we replace each tree in the double trees with a graph that has a complete binary tree as a spanning tree (the edge between the two graphs will be between the vertices at the roots of the spanning trees, and connections between copies of the "double graphs" in the crossproduct will be between corresponding vertices in the spanning trees). We could therefore construct a three-dimensional finite element mesh from our example that would represent the channel tunnel (the Chunnel) between England and France; the chunnel is a pair of long tubes with a small connection between the center.

References

[AG84] B. Aspvall and J. R. Gilbert. Graph coloring using eigenvalue decomposition. *SIAM Journal on Algebraic and Discrete Methods*, 5(4):526–538, December 1984.

[AGM87] N. Alon, Z. Galil, and V. D. Milman. Better expanders and superconcentrators. *Journal of Algorithms*, 8:337–347, 1987.

[Alo86] N. Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986.

[AM85] N. Alon and V. D. Milman. λ_1, isoperimetric inequalities for graphs, and superconcentrators. *Journal of Combinatorial Theory, Series B*, 38:73–88, 1985.

[Bar82] Earl R. Barnes. An algorithm for partitioning the nodes of a graph. *SIAM J. Alg. Disc. Meth.*, 3(4):541–550, December 1982.

[Bop87] R. Boppana. Eigenvalues and graph bisection: An average-case analysis. In *28th Annual Symposium on Foundations of Computer Science*, pages 280–285, Los Angeles, October 1987. IEEE.

[CDS79] D. M. Cvetković, M. Doob, and H. Sachs. *Spectra of Graphs.* Academic Press, New York, 1979.

[CGT94] Tony F. Chan, John R. Gilbert, and Shang-Hua Teng. Geometric spectral partitioning. Technical Report CSL-94-15, Xerox Parc, July 1994. Revised January 1995.

[DH73] W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM J. Res. Develop.*, 17:420–425, 1973.

[Fie73] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98):298–305, 1973.

[Fie75] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25(100):619–633, 1975.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP–completeness.* Freeman, San Francisco, 1979.

[GL89] G. H. Golub and C. F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, 1989.

[GMT95] John Gilbert, G.L. Miller, and Shang-Hua Teng. Geometric mesh partitioning: Implementation and experiments. In *9th International Parallel Processing Symposium*, Santa Barbara, April 1995. IEEE. to appear.

[HK92] Lars Hagen and Andrew B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEE Transactions on Computer-Aided Design*, 11(9):1074–1085, September 1992.

[LR88] T. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *29th Annual Symposium on the Foundations of Computer Science*, pages 422–431. IEEE Computer Society, October 1988.

[Moh88] B. Mohar. The laplacian spectrum of graphs. In *Sixth International Conference on the Theory and Applications of Graphs*, pages 871–898, 1988.

[Moh89] Bojan Mohar. Isoperimetric numbers of graphs. *Journal of Combinatorial Theory, Series B*, 47:274–291, 1989.

[MTV91] Gary L. Miller, Shang-Hua Teng, and Stephen A. Vavasis. A unified geometric approach to graph separators. In *32th Annual Symposium on Foundations of Computer Science*, pages 538–547, Puerto Rico, Oct 1991. IEEE.

[PSL90] Alex Pothen, Horst D. Simon, and Kang-Pu Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Anal. Appl.*, 11(3):430–452, July 1990.

[Sim91] H. D. Simon. Partitioning of unstructured problems for parallel processing. *Computing Systems in Engineering*, 2(2/3):135–148, 1991.

[Wil65] J. H. Wilkinson. *The Algebraic Eigenvalue Problem.* Oxford University Press, Oxford, 1965.

[YG73] D. M. Young and R. T. Gregory. *A Survey of Numerical Mathematics*, volume II of *Addison-Wesley Series in Mathematics*. Addison-Wesley, Reading, MA, 1973.